

## IT2024- USER INTERFACE DESIGN

### UNIT - I

#### Human-Computer Interface

- The need for people to communicate with each other has existed since we first walked upon this planet. The lowest and most common level of communication modes we share are movements and gestures independent, that is, they permit people who do not speak the same language to deal with one another.
- The next higher level, in terms of universality and complexity, is spoken language. Most people can speak one language, some two or more. A spoken language is a very efficient mode of communication if both parties to the communication understand it. At the third and highest level of complexity is written language. While most people speak, not all can write. But for those who can, writing is still nowhere near as efficient a means of communication as speaking.
- In modern times, we have the typewriter, another step upward in communication complexity. Significantly fewer people type than write. (While a practiced typist can find typing faster and more efficient than handwriting, the unskilled may not find this the case.) Spoken language, however, is still more efficient than typing, regardless of typing skill level.

#### ➤ **Introduction of the Graphical User Interface**

- Finally, in the 1970s, another dialog alternative surfaced. Research at Xerox's Palo Alto Research Center provided an alternative to the typewriter, an interface using a form of human gesturing, the most basic of all human communication methods. The Xerox systems, Altus and STAR, introduced the mouse and pointing and selecting as the primary human-computer communication method.
- The user simply pointed at the screen, using the mouse as an intermediary. These systems also introduced the graphical user interface as we know it today. Ivan

Sutherland at the Massachusetts Institute of Technology (MIT) is given credit for first introducing graphics with his Sketchpad program in 1963.

## Characteristics of Graphical and Web User Interfaces

- In brief, a graphical user interface can be defined as follows. A *user interface*, as recently described, is a collection of techniques and mechanisms to interact with something. In a *graphical* interface, the primary interaction mechanism is a pointing device of some kind.
- This device is the electronic equivalent to the human hand. What the user interacts with is a collection of elements referred to as *objects*. They can be seen, heard, touched, or otherwise perceived. Objects are always visible to the user and are used to perform tasks. They are interacted with as entities independent of all other objects. People perform operations, called *actions*, on objects. The operations include accessing and modifying objects by pointing, selecting, and manipulating. All objects have standard resulting behaviors.

## **The Concept of Direct Manipulation**

- The term used to describe this style of interaction for graphical systems was first used by Shneiderman (1982). He called them “direct manipulation” systems, suggesting that they possess the following characteristics:
  - **The system is portrayed as an extension of the real world.** It is assumed that a person is already familiar with the objects and actions in his or her environment of interest. The system simply replicates them and portrays them on a different medium, the screen.
  - A person has the power to access and modify these objects, among which are windows. A person is allowed to work in a familiar environment and in a familiar way, focusing on the data, not the application and tools. The physical organization of

the system, which most often is unfamiliar, is hidden from view and is not a distraction.

- **Continuous visibility of objects and actions.** Like one's desktop, objects are continuously visible. Reminders of actions to be performed are also obvious, labeled buttons replacing complex syntax and command names. Cursor action and motion occurs in physically obvious and intuitively natural ways.
- **Actions are rapid and incremental with visible display of results.** Since tactile feedback is not yet possible (as would occur with one's hand when one touches something) the results of actions are immediately displayed visually on the screen in their new and current form. Auditory feedback may also be provided. The impact of a previous action is quickly seen, and the evolution of tasks is continuous and effortless.
- **Incremental actions are easily reversible.** Finally, actions, if discovered to be incorrect or not desired, can be easily undone.
- **Earlier Direct Manipulation Systems** Using the above definition, the concept of direct manipulation actually preceded the first graphical system. The earliest full-screen text editors possessed similar characteristics. Screens of text resembling a piece of paper on one's desk could be created (extension of real world) and then reviewed in their entirety (continuous visibility). Editing or restructuring could be easily accomplished (through rapid incremental actions) and the results immediately seen. Actions could be reversed when necessary. It took the advent of graphical systems to crystallize the direct manipulation concept.

## The Web User Interface

- The expansion of the World Wide Web since the early 1990s has been truly amazing. Once simply a communication medium for scientists and researchers, its many and pervasive tentacles have spread deeply into businesses, organizations, and homes around the world. Unlike earlier text-based and GUI systems that were developed and nurtured in

an organization's Data Processing and Information Systems groups, the Web's roots were sown in a market-driven society thirsting for convenience and information.

- Web interface design is essentially the design of navigation and the presentation of information. It is about content, not data. Proper interface design is largely a matter of properly balancing the structure and relationships of menus, content, and other linked documents or graphics. The design goal is to build a hierarchy of menus and pages that feels natural, is well structured, is easy to use, and is truthful. The Web is a navigation environment where people move between pages of information, not an application environment. It is also a graphically rich environment.
- Web interface design is difficult for a number of reasons. First, its underlying design language, HTML, was never intended for creating screens to be used by the general population. Its scope of users was expected to be technical. HTML was limited in objects and interaction styles and did not provide a means for presenting information in the most effective way for people. Next, browser navigation retreated to the pre-GUI era.

## The Popularity of the Web

- While the introduction of the graphical user interface revolutionized the user interface, the Web has revolutionized computing. It allows millions of people scattered across the globe to communicate, access information, publish, and be heard. It allows people to control much of the display and the rendering of Web pages. Aspects such as typography and colors can be changed, graphics turned off, and decisions made whether or not to transmit certain data over nonsecure channels or whether to accept or refuse cookies. Nowhere in the history of computing has the user been given so much control. Web usage has reflected this popularity. The number of Internet hosts has risen dramatically.
- **Characteristics of a Web Interface**

- A Web interface possesses a number of characteristics, some similar to a GUI interface, and, as has already been shown, some different. In the following paragraphs many of these specific commonalities and differences will be examined. Also, the differing characteristics of printed page design and Web page design will be compared.
- **User focus.** *GUI* systems are about well-defined applications and data, about transactions and processes. Thorough attention must usually be addressed to tasks in need of completion. The *Web* is about information and navigation, an environment where people move back and forth in an unstructured way among many pages of information. Web use is most often characterized browsing and visual scanning of information to find what information is needed.
- **Data/information.** *GUI* data is typically created and used by known and trusted sources, people in the user's organization or reputable and reliable companies and organizations. The properties of the system's data are generally known, and the information is typically organized in an understandable and meaningful fashion. A notion of shared data exists, as does a notion of data privacy. The *Web* is full of unknown content typically placed there by others unknown to the user.
- **User tasks.** *GUI* system users install, configure, personalize, start, use, and upgrade programs. They open, use, and close data files. Fairly long times are spent within an individual publication, and people become familiar with many of its features and its design. *Web* users do things like linking to sites, browsing or reading pages, filling out forms, registering for services, participating in transactions, and downloading and saving things.
- **User's conceptual space.** In a *GUI* environment the user's conceptual space is controlled by the program and application. A user's access to data is constrained, and

space is made available where their data can be stored and managed. A *Web* user's space is infinite and generally unorganized. Little opportunity for meaningful organization of personal information exists.

- **Presentation elements.** The main presentation elements for *GUIs* are various kinds of windows, menus, controls, toolbars, messages, and data. Many elements are transient, dynamically appearing and disappearing based upon the current context of the interaction. They are also generally standardized as a result of the toolkits and style guides used. Elements are presented on screens exactly as specified by the designer. Common toolkits and industry conventions, however, are now being proposed.

### Characteristics of Graphical and Web User Interfaces

- **Context.**
- **Interaction.**
- **Response time.**
- **Visual style.**
- **System capability.**
- **Task efficiency.**
- **Consistency.**
- **User assistance.**

➤ **Integration.**

➤ **Security.**

## **Principles of User Interface Design**

An interface must really be just an extension of a person. This means that the system and its software must reflect a person's capabilities and respond to his or her specific needs. It should be useful, accomplishing some business objectives faster and more efficiently than the previously used method or tool did. It must also be easy to learn, for people want to do, not learn to do. Finally, the system must be easy and fun to use, evoking a sense of pleasure and accomplishment not tedium and frustration.

➤ **Principles for the Xerox STAR**

- The design of the Xerox STAR was guided by a set of principles that evolved over its lengthy development process (Smith, Harslem, Irby, Kimball, and Verplank, 1982; Verplank, 1988). These principles established the foundation for graphical interfaces.
- **The illusion of manipulable objects.** Displayed objects that are selectable and manipulable must be created. A design challenge is to invent a set of displayable objects that are represented meaningfully and appropriately for the intended application. It must be clear that these objects can be selected, and how to select them must be self-evident.
- **Visual order and viewer focus.** Attention must be drawn, at the proper time, to the important and relevant elements of the display. Effective visual contrast between various components of the screen is used to achieve this goal (STAR was

monochromatic so color was not used). Animation is also used to draw attention, as is sound. Feedback must also be provided to the user. Since the pointer is usually the focus of viewer attention, it is a useful mechanism for providing this feedback

- **Revealed structure.** The distance between one's intention and the effect must be minimized. Most often, the distance between intention and effect is lengthened as system power increases. The relationship between intention and effect must be tightened and made as apparent as possible to the user. The underlying structure is often revealed during the selection process. **Consistency.** Consistency aids learning. Consistency is provided in such areas as element location, grammar, font shapes, styles, and sizes, selection indicators, and contrast and emphasis techniques.
- **Appropriate effect or emotional impact.** The interface must provide the appropriate emotional effect for the product and its market. Is it a corporate, professional, and secure business system? Should it reflect the fantasy, wizardry, and bad puns of computer games?
- **A match with the medium.** The interface must also reflect the capabilities of the device on which it will be displayed. Quality of screen images will be greatly affected by a device's resolution and color-generation capabilities.

## General Principles

- The design goals in creating a user interface are described below. They are fundamental to the design and implementation of all effective interfaces, including GUI and Web ones. These principles are general characteristics of the interface, and they apply to all aspects. Specific guidelines on how to implement many of these goals will be presented in Part 2. The compilation is presented alphabetically, and the ordering is not intended to imply degree of importance. They are derived from the various principles described in Galitz (1992), IBM (1991, 2001), Mayhew (1992), Microsoft (1992, 1995, 2001), Open Software Foundation (1993), and Verplank (1988).



➤ ***Aesthetically Pleasing***

Provide visual appeal by following these presentation and graphic design principles:

- Provide meaningful contrast between screen elements.
- Create groupings.
- Align screen elements and groups.
- Provide three-dimensional representation.
- Use color and graphics effectively and simply.

## **UNIT – II**

### **The User Interface Design Process**

It presents an extensive series of guidelines for the interface design process. It is organized in the order of the development steps typically followed in creating a graphical system's or Web site's screens and pages.

### **Obstacles**

➤ Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observations about design:

- **Obstacles and Pitfalls in the Development Path**

Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observations about design: Nobody ever gets it right the first time. Development is hock-full of surprises. Good design requires living in a sea of changes. Making contracts to ignore change will never eliminate the need for change. Even if you have

made the best system humanly possible, people will still make mistakes when using it. Designers need good tools. You must have behavioral design goals like performance design goals. The first five conditions listed will occur naturally because people are people, both as users and as developers. These kinds of behavior must be understood and accepted in design. User mistakes, while they will always occur, can be reduced.

## Usability

- Bennett (1979) was the first to use the term *usability* to describe the effectiveness of human performance. In the following years a more formal definition was proposed by Shackel (1981) and modified by Bennett (1984). Finally, Shackel (1991) simply defined usability as “the capability to be used by humans easily and effectively, where, easily = to a specified level of subjective assessment, effectively = to a specified level of human performance.”
- **Usability Assessment in the Design Process** Usability assessment should begin in the early stages of the product development cycle and should be continually applied throughout the process. The assessment should
- **Common Usability Problems**
  - Mandel (1994) lists the 10 most common usability problems in graphical systems as reported by IBM usability specialists. They are:
    1. Ambiguous menus and icons.
    2. Languages that permit only single-direction movement through a system.
    3. Input and direct manipulation limits.
    4. Highlighting and selection limitations.
    5. Unclear step sequences.
    6. More steps to manage the interface than to perform tasks.

7. Complex linkage between and within applications.
8. Inadequate feedback and confirmation.
9. Lack of system anticipation and intelligence.
10. Inadequate error messages, help, tutorials, and documentation.

## **Important Human Characteristics in Design**

- We are complex organisms with a variety of attributes that have an important influence on interface and screen design. Of particular importance in design are perception, memory, visual acuity, foveal and peripheral vision, sensory storage, information processing, learning, skill, and individual differences.
- **Perception**
  - Proximity
  - Similarity
  - Matching patterns
  - Succinctness
  - Closure
  - Unity
  - Continuity
  - Balance
  - Expectancies
  - Context
  - Signals versus noise
- **Memory**
- **Sensory Storage**
- **Visual Acuity**
- **Foveal and Peripheral Vision**

- **Information Processing**
- **Mental Models**
- **Movement Control**
- **Learning**
- **Skill**
- **Individual Differences**

## **Human Interaction Speed**

- The speed at which people can perform using various communication methods has been studied by a number of researchers. The following, as summarized by Bailey (2000), have been found to be typical interaction speeds for various tasks. These speeds are also summarized
- **Reading.** The average adult, reading English prose in the United States, has a reading speed in the order of 250–300 words per minute. Proofreading text on paper has been found to occur at about 200 words per minute, on a computer monitor, about 180 words per minute (Ziefle, 1998). Nontraditional reading methods have also been explored in research laboratories.
- One technique that has dramatically increased reading speeds is called Rapid Serial Visual Presentation, or RSVP. In this technique single words are presented one at a time in the center of a screen. New words continually replace old words at a rate set by the reader. Bailey (1999a) tested this technique with a sample of people whose paper document reading speed was 342 words per minute. (With a speed range of 143 to 540 words per minute.) Single words were presented on a screen in sets at a speed sequentially varying ranging from 600 to 1,600 words per minute. After each set a comprehension test was administered.
- **Listening.** Words can be comfortably heard and understood at a rate of 150 to 160 words per minute. This is generally the recommended rate for audio books and video narration (Williams, 1998). Omoigui, et al, (1999) did find, however, that when normal speech is speeded up using compression, a speed of 210 words per minute results in no loss of comprehension.

- **Speaking.** Dictating to a computer occurs at a rate of about 105 words per minute (Karat, et al., 1999; Lewis, 1999). Speech recognizer misrecognitions often occur, however, and when word correction times are factored in, the speed drops significantly, to an average of 25 words per minute. Karat, et al. (1999) also found that the speaking rate of new users was 14 words per minute during transcription and 8 words per minute during composition.
- **Keying.** Fast typewriter typists can key at rates of 150 words per minute and higher. Average typing speed is considered to be about 60–70 words per minute.

## **Business Function**

### **Requirements Analysis**

- The objective of this phase is to establish the need for a system. A requirement is an objective that must be met. A product description is developed and refined, based on input from users or marketing. There are many techniques for capturing information for determining requirements. Keil and Carmel (1995), Popowicz (1995), and Fuccella et al. (1999) described many of the methods summarized and discussed shortly. They have also provided insights into their advantages and disadvantages.
- The techniques listed are classified as direct and indirect. Direct methods consist of face-to-face meetings with, or actual viewing of, users to solicit requirements. Indirect methods impose an intermediary, someone or something, between the users and the developers. Before beginning the analysis, the developer should be aware of the policies and work culture of the organization being studied. He or she should also be familiar with any current system or process the new system is intended to supplement or replace.

#### Individual Face-to-Face Interview

- A one-on-one visit with the user to obtain information. It may be structured or somewhat open-ended.

#### Telephone Interview or Survey

- A structured interview conducted via telephone.

#### Traditional Focus Group

- A small group of users and a moderator brought together to verbally discuss the requirements.

#### Facilitated Team Workshop

- A facilitated, structured workshop held with users to obtain requirements information. Similar to the Traditional Focus Group.

#### Observational Field Study

- Users are observed and monitored for an extended time to learn what they do.

#### Requirements Prototyping

- A demo, or very early prototype, is presented to users for comments concerning functionality.

#### User-Interface Prototyping

- A demo, or early prototype, is presented to users to uncover user-interface issues and problems.

#### Usability Laboratory Testing

- Users at work are observed, evaluated, and measured in a specially constructed laboratory.

#### Card Sorting for Web Sites

- A technique to establish groupings of information for Web sites.

#### MIS Intermediary

- A company representative defines the user's goals and needs to designers and developers.

#### Paper Survey or Questionnaire

- A survey or questionnaire is administered to a sample of users using traditional mail methods to obtain their needs.

#### Electronic Survey or Questionnaire

- A survey or questionnaire is administered to a sample of users using e-mail or the Web to obtain their needs.

#### Electronic Focus Group

- A small group of users and a moderator discuss the requirements online using workstations.

#### Marketing and Sales

- Company representatives who regularly meet customers obtain suggestions or needs, current and potential.

#### Support Line

- Information collected by the unit that helps customers with day-to-day problems is analyzed (Customer Support, Technical Support, Help Desk, etc.).

#### E-Mail or Bulletin Board

- Problems, questions, and suggestions from users posted to a bulletin board or through e-mail are analyzed.

#### User Group

- Improvements are suggested by customer groups who convene periodically to discuss software usage.

#### Competitor Analyses

- A review of competitor's products or Web sites is used to gather ideas, uncover design requirements and identify tasks.

#### Trade Show

- Customers at a trade show are presented a mock-up or prototype and asked for comments.

#### Other Media Analysis

- An analysis of how other media, print or broadcast, present the process, information, or subject matter of interest.

#### System Testing

- New requirements and feedback are obtained from ongoing product testing

## Direct Methods

- The significant advantage of the direct methods is the opportunity they provide to hear the user's comments in person and firsthand. Person-to-person encounters permit multiple channels of communication (body language, voice inflections, and so on) and provide the opportunity to immediately follow up on vague or incomplete data. Here are some recommended direct methods for getting input from users.

- ***Individual Face-to-Face Interview***

A one-on-one visit is held with the user. It may be structured or more open-ended. The interview must have focus and topics to be covered must be carefully planned so data is collected in a common framework, and to ensure that all important aspects are thoroughly covered.

- A formal questionnaire should not be used, however. Useful topics to ask the user to describe in an interview include: The activities performed in completing a task or achieving a goal or objective.
- The methods used to perform an activity.
- What interactions exist with other people or systems.
- It is also very useful to also uncover any: Potential measures of system usability
- Unmentioned exceptions to standard policies or procedures.



- Relevant knowledge the user must possess to perform the activity.
- **.Telephone Interview or Survey**
- **Traditional Focus Group**
- **Facilitated Team Workshop**
- **Observational Field Study**
- **Requirements Prototyping**
- **User-Interface Prototyping**
- **Usability Laboratory Testing**
- **Card Sorting for Web Sites**

## **Indirect Methods**

- An indirect method of requirements determination is one that places an intermediary between the developer and the user. This intermediary may be electronic or another person. Using an intermediary can certainly provide useful information. Working through an intermediary, however, takes away the multichannel communication advantages of face-to-face user-developer contact. Some electronic intermediaries do provide some advantages, as will be described. Imposition of a human intermediary can also create these additional problems. First, there may be a filtering or distortion of the message, either intentional or unintentional. Next, the intermediary may not possess a complete, or current, understanding of user's needs, passing on an incomplete or incorrect message. Finally, the intermediary may be a mechanism that discourages direct user-developer contact for political reasons. Indirect methods include the following.

- **MIS Intermediary**
- **Paper Survey or Questionnaire**
- **Electronic Survey or Questionnaire**
- **Electronic Focus Group Similar**
- **Marketing and Sales**
- **Support Line**

- **E-Mail, Bulletin Boards or Guest Book**
- **User Group**
- **Competitor Analysis**
- **Trade Show**
- **Other Media Analysis**
- **System Testing**

## **Basic Business Functions**

- A detailed description of what the product will do is prepared. Major system functions are listed and described, including critical system inputs and outputs. A flowchart of major functions is developed. The process the developer will use is summarized as follows:

- Gain a complete understanding of the user's mental model based upon:
  - The user's needs and the user's profile.
  - A user task analysis.
  - Develop a conceptual model of the system based upon the user's mental model.
- This includes:
  - Defining objects.
  - Developing metaphors.

➤ **Understanding the User's Mental Model**

The next phase in interface design is to thoroughly describe the expected system user or users and their current tasks. The former will be derived from the kinds of information collected in Step 1 "Understand the User or Client," and the requirements analysis techniques described above. A goal of task analysis, and a goal of understanding the user, is to gain a picture of the user's mental model. A mental model is an internal representation of a person's current conceptualization and understanding of something. Mental models are gradually developed in order to understand, explain, and do something. Mental models enable a person to predict the actions necessary to do things if the actions have been forgotten or have not yet been encountered.

➤ **Developing Conceptual Models**

The output of the task analysis is the creation, by the designer, of a conceptual model for the user interface. A conceptual model is the general conceptual framework through which the system's functions are presented. Such a model describes how the interface will present objects, the relationships between objects, the properties of objects, and the actions that will be performed. A conceptual model is based on the user's mental model. Since the term mental model refers to a person's current level of knowledge about something, people will *always* have them. Since mental models are influenced by a person's experiences, and people have different experiences, no two user mental models are likely to be exactly the same. Each person looks at the interface from a slightly different perspective.

➤ **The User's New Mental Model**

When the system is implemented, and a person interacts with the new system and its interface, an attempt will be made by the person to understand the system based upon the existing mental model brought to the interaction. If the designer has correctly reflected the user's mental model in design, the user's mental model is reinforced and a feeling that the interface is intuitive will likely develop. Continued interaction with the system may influence and modify the user's concept of the system, and his or her mental model may be modified as well. Refinement of this mental model, a normal process, is aided by well-defined distinctions between objects and being consistent across all aspects of the interface. What happens, however, if the new system does not accurately reflect the user's existing mental model? The results include breakdowns in understanding, confusion, errors, loss of trust, and frustration. Another result is an inability to perform the task or job. Historically, when system designers have known in advance there was a gap between their conceptual model and the mental model the user would bring to the new system, designers have tried to bridge this gap through extensive documentation and training. The problems with this approach are: people are unproductive while being trained, people rarely read the documentation and training materials, and, even if the training material is read, the material is presented out of context. This creates difficulties for the users in understanding the material's relevance to their needs and goals.

## Design Standards

- A design standard or style guide documents an agreed-upon way of doing something. In interface design it describes the appearance and behavior of the interface and provides some guidance on the proper use of system components. It also defines the interface standards, rules, guidelines, and conventions that must be followed in detailed design. It will be based on the characteristics of the system's hardware and software, the principles of good interface and screen design, the needs of system users, and any unique company or organization requirements that may exist.

- **Value of Standards and Guidelines**

Developing and applying design standards or guidelines achieves design consistency.

This is valuable to *users* because the standards and guidelines:

- Allow faster performance.
- Reduce errors.
- Reduce training time.
- Foster better system utilization.
- Improve satisfaction.
- Improve system acceptance.
- They are valuable to system *developers* because they:
  - Increase visibility of the human-computer interface.
  - Simplify design.
  - Provide more programming and design aids, reducing programming time.
  - Reduce redundant effort.
  - Reduce training time.
  - Provide a benchmark for quality control testing.

## System Timings

- Training and documentation are also an integral part of any development effort. System training will be based on user needs, system conceptual design, system learning goals, and system performance goals.
- Training may include such tools as formal or video training, manuals, online tutorials, reference manuals, quick reference guides, and online help. (Various types of training methods are more fully addressed.
- Training needs must be established and training components developed as the design process unfolds. This will ensure that the proper kinds of training are defined, properly integrated with the design, and developed correctly.
- This will also assure that the design is not imposing an unreasonable learning and training requirement on the user.
- Any potential problems can also be identified and addressed earlier in the design process, reducing later problems and modification costs.

## Human Considerations in Screen Design

- How to Distract the Screen User
- What Screen Users
- Want What Screen Users Do
- Interface Design Goals
- The Test for a Good Design
- Screen Meaning and Purpose
- Organizing Screen Elements Clearly and Meaningfully
- Consistency

- Ordering of Screen Data and Content
- Upper-Left Starting Point
- Screen Navigation and Flow
- Visually Pleasing Composition
- Amount of Information
- Distinctiveness
- Focus and Emphasis
- Conveying Depth of Levels or a Three-Dimensional Appearance
- Presenting Information Simply and Meaningfully
- Reading, Browsing, and Searching on the Web
- Intranet Design Guidelines
- Extranet Design Guidelines
- Statistical Graphics

## Structures of Menus

- Single Menu
- Sequential Linear Menus
- Simultaneous Menus
- Hierarchical Menus
- Connected Menus
- Event-Trapping Menus

## Functions of Menus

- Navigation to a New Menu
- Execute an Action or Procedure

- Displaying Information
- Data or Parameter Input

## **Content of Menus**

- Menu Context
- Menu Title
- Choice Descriptions
- Completion Instructions

## **Formatting of Menus**

- Consistency
- Display
- Presentation
- Organization
- Complexity
- Item Arrangement
- Ordering
- Groupings

## **Phrasing the Menu**

- Menu Titles
- Menu Choice Descriptions
- Menu Instructions
- Intent Indicators

- Keyboard Equivalents
- Keyboard Accelerators

## Selecting Menu Choices

- Initial Cursor Positioning
- Choice Selection
- Defaults
- Unavailable Choices
- Mark Toggles or Settings
- Toggled Menu Items

## Navigating Menus

- Web Site Navigation Problems
- Navigation Goals
- Web Site Navigation
- Components of a Web Navigation System
- Web Site Navigation Guidelines
- Maintaining a Sense of Place



## Graphical Menus

- Menu Bar
- Pull-Down Menu
- Cascading Menus
- Pop-up Menus
- Tear-off Menus
- Iconic Menus
- Pie Menus
- Default Menu Items

<http://csetube.tk/>

## UNIT – III

### Characteristics

- A window is seen to possess the following characteristics:
- A name or title, allowing it to be identified.
  - A size in height and width (which can vary).
  - A state, accessible or active, or not accessible. (Only active windows can have their contents altered.)
  - Visibility—the portion that can be seen. (A window may be partially or fully hidden behind another window, or the information within a window may extend beyond the window's display area.)
  - A location, relative to the display boundary.
  - Presentation, that is, its arrangement in relation to other windows. It may be tiled, overlapping, or cascading.
  - Management capabilities, methods for manipulation of the window on the screen.
  - Its highlight, that is, the part that is selected.
  - The function, task, or application to which it is dedicated.

### **Components**

- Frame
- Title Bar
- Title Bar Icon
- Window Sizing Buttons
- Menu Bar
- Status Bar
- Scroll Bars
- Split Box
- Toolbar
- Command Area
- Size Grip
- Work Area

## Presentation Styles

- Tiled Windows
- Overlapping Windows
- Cascading Windows
- Picking a Presentation Style
  - Use tiled windows for:
    - Single-task activities.
    - Data that needs to be seen simultaneously.
    - Tasks requiring little window manipulation.
    - Novice or inexperienced users.
  - Use overlapping windows for:
    - Switching between tasks.
    - Tasks necessitating a greater amount of window manipulation.

## Types

- Primary Window
- Secondary Windows
  - Modal and Modeless
  - Cascading and Unfolding
- Dialog Boxes
- Property Sheets and Property Inspectors
- Message Boxes
- Palette Windows
- Pop-up Windows

## Management

- Single-Document Interface
- Multiple-Document Interface
- Workbooks
- Projects

## Organizations

- Organize windows to support user tasks.
- Support the most common tasks in the most efficient sequence of steps.
- Use primary windows to:
  - Begin an interaction and provide a top-level context for dependent windows.
  - Perform a major interaction.
- Use secondary windows to:
  - Extend the interaction.

- Obtain or display supplemental information related to the primary window.
- Use dialog boxes for:
  - Infrequently used or needed information.
  - “Nice-to-know” information.

## Operations

- Active Window
- Opening a Window
- Sizing Windows
- Window Placement
- Window Separation
- Moving a Window
- Resizing a Window
- Other Operations
- Window Shuffling
- Keyboard Control/Mouse less Operation
- Closing a Window

## Web Systems

➤ Web systems have limited windowing capabilities. The *frame* concept does provide window-like ability, and JavaScript does provide *pop-up* windows.

### ➤ Frames

#### ■ Description:

- Multiple Web screen panes that permit the displaying of multiple documents on a page.
- These documents can be independently viewed, scrolled, and updated.

- The documents are presented in a tiled format.

- Proper usage:

- For content expected to change frequently.
- To allow users to change partial screen content.
- To permit users to compare multiple pieces of information.

- Guidelines:

- Use only a few frames (three or less) at a given time.
- Choose sizes based upon the type of information to be presented.
- Never force viewers to resize frames to see information.
- Never use more than one scrolling region on a page.

➤ **Description.**

The Web is, historically, essentially a single page (or, by analogy, a single window) entity. While providing significant interface benefits, it is also a reversal of the interface evolution process that led from single-screen technology to windowing. To counteract this shortcoming, *frames* were created. A frame is an independent pane of information presented in a Web page, or, again by analogy, as multiple windows. Frames, however, are presented as tiled, with no overlapping capability. The interaction richness, support, and contextual cues provided by overlapping windows are lacking. Frames, then, allow the displaying of multiple documents on a single Web page. These multiple documents can be independently viewed, scrolled, and updated.

➤ **Proper usage.**

Frames are useful in situations where portions of the page content are expected to change frequently. The volatile information can be separated from other page content and placed within a frame, thereby requiring only a portion of the page's content to be modified. Frames are also useful for allowing the user to change page content; navigation links can be placed in one frame and the resulting content displayed within another frame. As different links are selected, the content in the related frame changes. Finally, frames more

effectively allow users to compare multiple pieces of related information displayed within the different frames.

## Device-Based Controls – Characteristics

- Several specific tasks are performed using graphical systems.
  - To point at an object on the screen.
  - To select the object or identify it as the focus of attention.
  - To drag an object across the screen.
  - To draw something free form on the screen.
  - To track or follow a moving object.
  - To orient or position an object.
  - To enter or manipulate data or information
- Trackball
- Joystick
- Graphic Tablet
- Touch Screen
- Light Pen
- Voice
- Mouse
- Keyboard

## Screen- Based Controls

### Operable Controls

➤ Operable controls are those that permit the entry, selection, changing, or editing of a particular value, or cause a command to be performed. Classes include buttons, text entry/read-only, selection, combination entry/selection, and other specialized controls.

#### ➤ **Buttons**

##### ■ Description:

- A square or rectangular-shaped control with a label inside that indicates action to be accomplished.
- The label may consist of text, graphics, or both.

##### ■ Purpose:

- To start actions.
- To change properties.
- To display a pop-up menu.

##### ■ Advantages:

- Always visible, reminding one of the choices available.
- Convenient.
- Can be logically organized in the work area.
- Can provide meaningful descriptions of the actions that will be performed.
- Larger size generally provides faster selection target.
- Can possess 3-D appearance:



- Adds an aesthetically pleasing style to the screen.
- Provides visual feedback through button movement when activated.
- May permit use of keyboard equivalents and accelerators.
- Faster than using a two-step menu bar/pull-down sequence.
- Disadvantages:
  - Consumes screen space.
  - Size limits the number that may be displayed.
  - Requires looking away from main working area to activate.
  - Requires moving the pointer to select.
- Proper usage:
  - Use for frequently used actions that are specific to a window.
    - To cause something to happen immediately.
    - To display another window.
    - To display a menu of options.
    - To set a mode or property value.

## Text Boxes

- Description:
  - A control, usually rectangular in shape, in which:
    - Text may be entered or edited.
    - Text may be displayed for read-only purposes.
  - Usually possesses a caption describing the kind of information contained within it.
  - An outline field border:
    - Is included for enterable/editable text boxes.
    - Is not included for read-only text boxes.
  - Two types exist:
    - Single line.
    - Multiple line.
  - When first displayed, the box may be blank or contain an initial value.

■ Purpose:

- To permit the display, entering, or editing of textual information.
- To display read-only information.

■ Advantages:

- Very flexible.
- Familiar.
- Consumes little screen space.

■ Disadvantages:

- Requires use of typewriter keyboard.
- Requires user to remember what must be keyed.

■ Proper usage:

- Most useful for data that is:
  - Unlimited in scope.
  - Difficult to categorize.
  - Of a variety of different lengths.
- When using a selection list is not possible.

## Selection Controls

- Radio Buttons
- Check Boxes
- Palettes
- List Boxes
- List View Controls
- Drop – down / Pop-up List Boxes

## Combination / Selection Controls

- Spin Boxes

- Combo Boxes
- Drop – down / Pop-up Combo Boxes

## **Custom Controls**

- Implement custom controls with caution.
- If used, make the look and behavior of custom controls different from that of standard controls.
- Many toolkits and interface builders provide the ability to create custom controls; implement them with caution. The user is currently presented with a multitude of controls whose usage and operation must be learned and remembered.
- The addition of custom controls adds to this learning and increases system complexity. If custom controls must be developed and implemented, make their look and behavior as different as possible from the standard controls. This will avoid confusion between the various controls.

## **Presentation Controls**

- Static Text Fields
- Group Boxes
- Column Headings
- ToolTips
- Balloon Tips
- Progress Indicators
- Sample Box

- Scrolling Tickers

## UNIT – IV

### Text For Web Pages

**Words, Sentences, Messages, and Text** - All communications should simply, clearly, and politely provide the information one must have to effectively use a system.

**Readability** - The degree to which prose can be understood, based upon the complexity of its words and sentences determined by many factors, including word length, word commonality, sentence length, and the number of syllables in a sentence

#### **Choosing the Proper Words**

- Do not use
  - Jargon, words, or terms
    - Unique to the computer profession.
    - With different meanings outside of the computer profession.
    - Made up to describe special functions or conditions.
  - Abbreviations or acronyms
    - Unless the abbreviation or acronym is as familiar as a full word or phrase.
  - Word contractions, suffixes, and prefixes.
- Use
  - Short, familiar words.
  - Standard alphabetic characters.
  - Complete words.
  - Positive terms.
  - Simple action words; avoid noun strings.
  - The “more” dimension when comparing.
  - Consistent words.
- Do not
  - Stack words.
  - Hyphenate words.
    - Include punctuation for abbreviations, mnemonics, and acronyms.

#### **Writing Sentences and Messages**

- Sentences and messages must be

- Brief and simple.
- Limited to no more than twenty words per sentence.
- Limited to no more than six sentences per paragraph.
- Written at an eighth grade level or less for the general population.
- Directly and immediately usable.
- An affirmative statement.
- In an active voice.
- In the temporal sequence of events.
- Structured so that the main topic is near the beginning.
- Of parallel construction.
- Sentences and messages must be of the proper tone:
  - Nonauthoritarian.
  - Nonthreatening.
  - Nonanthropomorphic.
  - Nonpatronizing.
  - Nonpunishing.
  - Cautious in the use of humor

### **Kinds of Messages**

1. System messages
2. Status messages.
3. Informational messages
4. Warning messages.
5. Critical messages
6. Question messages.

### **Message Box Controls**

- Command buttons:
  - If a message requires no choices to be made but only acknowledgment,
    - Include an OK button.
  - If a message requires a choice to be made, provide a command button for each option.
    - Include OK and Cancel buttons only when the user has the option of continuing or stopping the action.
    - Include Yes and No buttons when the user must decide how to continue.
    - If these choices are too ambiguous, label the command buttons with the names of specific actions.
  - If a message allows initiation of an action to correct the situation described,
    - Include a properly labeled button initiating the corrective action.
  - If a message describes an interrupted process whose state cannot be restored,
    - Provide a Stop button.
  - If a message offers an opportunity to cancel a process as well as to perform or not perform an action,
    - Provide a Cancel button.
  - If more details about a message topic must be presented,
    - Provide a Help button.

- Designate the most frequent or least destructive option as the default.
- Display a message box only when the window of an application is active.
- Display only one message box for a specific condition.
- Close box:
  - Enable the title bar Close box only if the message includes a Cancel button

### ***Instructional Messages***

- Provide instructional information at the depth of detail needed by the user.
- Locate it at strategic positions on the screen.
- Display it in a manner that visually differentiates it from other screen elements.
- In writing, follow all relevant writing guidelines for words, sentences, and messages.

### ***Presenting Text***

- Display prose text in mixed upper- and lower-case letters.
- Fonts:
  - Use plain and simple fonts.
  - Choose a minimum point size of 12 to 14.
  - Use proportional fonts.
- Justification:
  - Left-justify.
  - Do not right-justify.
  - Do not hyphenate words.
- Line Length:
  - If fast reading is required, use line lengths of about 75-100 characters.
    - If user preference is important, a length of about 50-60 characters is acceptable.
    - Very narrow columns should be avoided.
- Line endings:
  - Coincide with grammatical boundaries.
- Line spacing:
  - Increase line spacing to enhance legibility and readability.
- Content:
  - Use headings to introduce a new topic.
  - Make first sentences descriptive.
  - Separate paragraphs by at least one blank line.
  - Start a fresh topic on a new page.
  - Use lists to present facts.
  - Emphasize important things by
    - Positioning.
    - Boxes.
    - Bold typefaces.
    - Indented margins.
- Provide a screen design philosophy consistent with other parts of the system.

## **Effective Feedback**

- To be effective, feedback to the user for an action must occur within certain time limits.

### **Response Time**

- System responsiveness should match the speed and flow of human thought processes.
  - If continuity of thinking is required and information must be remembered throughout several responses, response time should be less than 1 or 2 seconds.
  - If human task closures exist, high levels of concentration are not necessary, and moderate short-term memory requirements are imposed; response times of 2 to 4 seconds are acceptable.
  - If major task closures exist, minimal short-term memory requirements are imposed; responses within 4 to 15 seconds are acceptable.
  - When the user is free to do other things and return when convenient, response time can be greater than 15 seconds.
- Constant delays are preferable to variable delays

### **Web Page Download Times**

1. Longer delay times are tolerated by
2. Novice users (Shneiderman, 1998; Ryan and Valverde, 2003 ).
3. Older users (Selvidge, 2003).
4. People not experienced with high bandwidth connections (Selvidge, 2003).
5. New visitors to a site (Bouch et al., 2000).
6. People performing important tasks (Ceaparu et al., 2004).
7. People performing successful tasks (Watts-Perotti and Woods, 1999).
8. Better content (Ryan and Valverde, 2003).
9. Perceived delays in presentation cause sites to be evaluated as
10. Less interesting (Ramsay et al., 1998).
11. Possessing lower quality content (Sears and Jacko, 2000).
12. Interfering with task continuity and use flow (Shubin and Meehan, 1997).
13. Less rememberable (Shubin and Meehan, 1997).
14. Less secure (Bouch et al., 2000).
15. Exceedingly slow sites can lead people to believe that they made an error (Lazar and Norico, 2000).

### **Page Downloading**

- Maximum downloading time should be about 10 seconds.
  - Use incremental or progressive image presentation.
- Images should be
  - Small.
  - Lean (Low bit depth, resolution, use of fewer colors, and so forth).
  - Reusable.
  - Contain pixel height and width dimensions in a tag.
- Use thumbnail images to preview larger images.
- Use simple background images.

### **Dealing with Time Delays**

- Button click acknowledgement:
  - Acknowledge all button clicks by visual or aural feedback within one-tenth of a second.
- Waits of up to 10 seconds:
  - If an operation takes 10 seconds or less to complete, present a “busy” signal until the operation is complete.
    - Display, for example, an animated hourglass pointer.
- Waits of 10 seconds to 1 minute:
  - If an operation takes longer than 10 seconds to complete, display
    - A rolling barber’s pole or other large animated object.
    - Additionally, a progress indicator, percent complete message, or elapsed time message.
- Waits over 1 minute:
  - Present an estimate of the length of the wait.
  - Display a progress indicator, percent complete message, or elapsed time message.
  - When a long operation is completed, present an acknowledgment that it is completed.
    - A significantly changed screen appearance.
    - An auditory tone.
  - If an operation is very time-consuming
    - Consider breaking the operation into subtasks and providing progress indicators for each subtask.
    - Allow users to start a new activity while waiting.
- Long, invisible operations:
  - When an operation not visible to the user is completed, present an acknowledgment that it is completed.
    - A message.
    - An auditory tone.
- Progress indicator:



- Along rectangular bar that is initially empty but filled as the operation proceeds.
  - Dynamically fill the bar.
  - Fill it with a color or shade of gray.
  - Fill it from left to right or bottom to top.
- Percent complete message:
  - A message that indicates the percent of the operation that is complete.
  - Useful if a progress indicator takes too long to update.
- Elapsed time message:
  - A message that shows the amount of elapsed time that the operation is consuming.
  - Useful if
    - The length of the operation is not known in advance.
    - A particular part of the operation will take an unusually long time to complete.
- Web page downloads:
  - For pages requiring download times greater than 5 seconds, give the user something to do while waiting.
    - Quickly present, at the top of the downloading page, an image, text or links.
  - Advise of long download times.
  - Warn of “Time Outs.”

## **Guidance and Assistance**

- Prevention:
  - Disable inapplicable choices.
  - Use selection instead of entry controls.
  - Use aided entry.
  - Accept common misspellings, whenever possible.
  - Before an action is performed,
    - Permit it to be reviewed.
    - Permit it to be changed or undone.
  - Provide a common action mechanism.
  - Force confirmation of destructive actions.
    - Let expert users disable less critical confirmations.
  - Provide an automatic and continuous Save function.
- Detection:
  - For conversational dialogs, validate entries as close to point of entry as possible
    - At character level.
    - At control level.
    - When the transaction is completed or the window closed.
  - For high speed, head-down data entry
    - When the transaction is completed or the window closed.
  - Leave window open.
  - Maintain the item in error on the screen.
  - Visually highlight the item in error.
  - Display an error message in a window.
    - Do not obscure item in error.

- Handle errors as gracefully as possible.
  - The greater the error, the more dramatic should be the warning.
- Use auditory signals conservatively.
- **Correction:**
- Preserve as much of the user's work as possible.
- At window-level validation, use a modeless dialog box to display an error list.
  - Highlight first error in the list.
  - Place cursor at first control with error.
  - Permit fixing one error and continuing to next error.
- Always give a person something to do when an error occurs.
  - Something to enter/save/reverse.
  - A Help button.
  - Someone to call.
- Provide a constructive correction message saying
  - What problem was detected.
  - Which items are in error.
  - What corrective action is necessary.
- Initiate a clarification dialog if necessary

### ***Help Facility Guidelines***

- **Kind:**
- Collect data to determine what types of Help are needed.
- **Training:**
- Inform users of availability and purpose of Help.
- **Availability:**
- Provide availability throughout the dialog.
- If no Help is available for a specific situation, inform the user of this and provide directions to where relevant Help may exist.
- **Structure:**
- Make them as specific as possible.
- Provide a hierarchical framework.
  - Brief operational definitions and input rules.
  - Summary explanations in text.
- **Interaction:**
- Provide easy accessibility.
- Leave the Help displayed until
  - The user exits.
  - The action eliminating the need for Help is performed.
- Provide instructions for exiting.
- Return to original position in dialog when Help is completed.
- **Location:**
- Minimize the obscuring of screen content.
- If in a window, position priorities are right, left, above, and below.
- **Content:**
- Define unfamiliar terms.
- Minimize Help's length.

- Develop modular dialogs that can be used to describe similar and dissimilar procedural elements of the interface.
- Provide step-by-step interface procedures to assist the user with specific problems.
- Provide procedural demonstrations of interface procedures to aid quick learning of simple operations.
- Provide information to help users select between multiple interface methods.
- Provide users with an understanding of representative tasks to increase their knowledge of when to apply specific skills.
- **Style:**
  - Provide easy browsing and a distinctive format.
    - Contents screens and indexes.
    - Screen headings and subheadings.
    - Location indicators.
    - Descriptive words in the margin.
    - Visual differentiation of screen components.
    - Emphasized critical information.
  - Use concise, familiar, action-oriented wording.
  - Refer to other materials, when necessary.
  - Never use Help to compensate for poor interface design.
- **Consistency:**
  - Provide a design philosophy consistent with other parts of the system.
- **Title:**
  - Place the word “Help” in all Help screen titles. Typical task-oriented examples.

## **Internationalization**

### **Localization**

- **When to do it:**
  - When the market includes few or no English speakers.
  - When translation is required by law or by custom.
  - When the widest possible market is desired.
- **When not to do it:**
  - When the audience already reads English.
- When the cost of retrofitting or rewriting the software is prohibitive.

### **Cultural Considerations**

*Power distance* is the degree of acceptance of inequality of power and authority. The greater the existing distance between the powerful and pedestrian, the greater the power distance.

*Individualism (vs. collectivism)* reflects how well integrated a person is into the larger group.

*Masculinity (vs. femininity)* is the balance between assertiveness or toughness and supportiveness or caring.

*Uncertainty avoidance* is the level of discomfort with unstructured and potentially unpredictable conditions.

*Long-term orientation (vs. short term)* is the trade-off between long-term reward and immediate recognition.

## **Words and Text**

- Use very simple English.
  - Develop a restricted vocabulary.
  - Restrict the sentence structure using noun-verb-object.
- Avoid
  - Acronyms and abbreviations.
  - Slang or obscure phrasing.
  - Stringing three nouns together.
  - Local or computer jargon.
  - A telegraphic writing style.
  - An over-friendly writing style.
  - Culturally specific examples.
  - References to national, racial, religious, and sexist stereotypes.
- Adhere to local user language idioms and cultural contexts.
- Keep the original term for words that cannot be translated.
- Allow additional screen space for the translation.
  - Horizontally, using Table 10.1.
  - Vertically.
- When translating to other languages, first do
  - European: German.
  - Middle East: Arabic.
  - Far East: Japanese.
- Position icon captions outside of the graphic.
- Modify mnemonics for keyboard access.
- Adhere to local formats for date, time, money, measurements, addresses, and telephone numbers.

## **Images and Symbols**

- Adhere to local cultural and social norms.
- Use internationally accepted symbols.
- Develop generic images.
- Be particularly careful with
  - Religious symbols (crosses and stars).
  - The human body.
  - Women.
  - Hand gestures.
  - Flags.
  - Controversial geographic maps.
  - The cross and check for check boxes.
- Review proposed graphical images early in the design cycle.

## **Color, Sequence, and Functionality**

- Adhere to local color connotations and conventions.
- Provide the proper information sequence.

- Provide the proper functionality.
- Remove all references to features not supported.

## **Accessibility**

It means a system must be designed to be usable by an almost unlimited range of people, essentially anyone who desires to use it.

Design objectives in creating accessibility for users with disabilities are

- Minimize all barriers that make a system difficult, or impossible, to use.
- Provide compatibility with installed accessibility utilities.

### **Accessibility Design**

- Consider accessibility issues during system planning, design, and testing.
- Provide compatibility with installed accessibility utilities.
- Provide a customizable interface.
- Follow standard Windows conventions.
- Use standard Windows controls.
- Assure online forms can be easily completed.

### **Visual Disabilities**

- Utilities:
  - Ensure compatibility with screen-review utilities.
  - Ensure compatibility with screen-enlargement utilities.
- Screen components:
  - Include meaningful screen and window titles.
  - Provide associated captions or labels for all controls, objects, icons, and graphics.
  - Include graphical menu choices.
  - Provide a textual summary for each statistical graphic.
  - Allow for screen element scalability.
  - Support system settings for high contrast for all user interface controls and client area content.
  - When a high contrast setting is established, hide any images drawn behind the text to maintain screen information legibility.
  - Avoid displaying or hiding information based on the movement of the pointer.
  - Exception: Unless it's part of the standard interface (example: ToolTips).
- Keyboard:
  - Provide a complete keyboard interface.
  - Provide a logical order of screen navigation.
- Color:
  - Use color as an enhancing design characteristic.
  - If used
    - Select color combinations that can be discriminated.
    - Ensure that the lightness contrast between foreground and background color is high.
    - Increase the lightness contrast between colors in the color spectrum (blues and reds).
    - Avoid combining dark colors from the middle of the spectrum with light colors from either end of the spectrum.

- Create the color combinations based on the system colors for window components.
- Do not define specific colors.
- Use tools to verify what colors will look like when seen by color-deficient people.

### ***Hearing Disabilities***

- Provide captions or transcripts of important audio content.
- Provide an option to display a visual cue for all audio alerts.
- Provide an option to adjust the volume.
- Use audio as an enhancing design characteristic.
- Provide a spell-check or grammar-check utility.

### ***Physical Movement Disabilities***

- Provide voice-input systems.
- Provide a complete and simple keyboard interface.
- Provide a simple mouse interface.
- Provide on-screen keyboards.
- Provide keyboard filters.

### ***Speech or Language Disabilities***

- Provide a spell-check or grammar-check utility.
- Limit the use of time-based interfaces.
- Never briefly display critical feedback or messages and then automatically remove them.
- Provide an option to permit the user to adjust the length of the time-out.

### ***Cognitive Disabilities***

- Permit modification and simplification of the interface.
- Limit the use of time-based interfaces.
- Do not briefly display critical feedback or messages and then automatically remove them.
- Provide an option to permit the user to adjust the length of the time-out.

### ***Seizure Disorders***

- Use elements that do not blink or flicker at rates between frequency ranges of 2 Hz and 55 Hz.
- Minimize the area of the screen that is flashing.
- Avoid flashing that has a high level of contrast between states.
- Provide an option to enable users to slow down or disable screen flashing.

### ***Usability for Lower-Literacy People***

- To aid lower-literacy Web users,
- Prioritize information.
- Avoid moving or changing text.
- Streamline the page design.
- Simplify navigation.
- Optimize search

### ***Documentation***

- Provide documentation on all accessible features.
- Provide documentation in alternate formats.
- Provide online documentation for people who have difficulty reading or handling printed material.

## **Icons**

pictorial images most often used to represent objects and actions with which users can interact with or that they can manipulate  
definition for icon kinds.

- **Resemblance** — An image that looks like what it means.
- **Symbolic** — An abstract image representing something.
- **Exemplar** — An image illustrating an example or characteristic of something.
- **Arbitrary** — An image completely arbitrary in appearance whose meaning must be learned.
- **Analogy** — An image physically or semantically associated with something.

### Characteristics of Icons

An icon possesses the technical qualities of syntactics, semantics, and pragmatics

- Provide icons that are
  - Familiar.
  - Clear and Legible.
  - Simple.
  - Consistent.
  - Direct.
  - Efficient.
  - Discriminable.
- Also consider the
  - Context in which the icon is used.
  - Expectancies of users.
  - Complexity of task.

### A Successful Icon

- Looks different from all other icons.
- Is obvious what it does or represents.
- Is recognizable when no larger than 16 pixels square.
- Looks as good in black and white as in color.

### Size

- Supply in all standard sizes.
  - $16 \times 16$  pixels
    - 16- and 256-color versions.
  - $32 \times 32$  pixels
    - 16- and 256-color versions.
    - Effective:  $24 \times 24$  or  $26 \times 26$  in  $32 \times 32$  icon.
  - $48 \times 48$  pixels
    - 16- and 256-color versions.
- Use colors from the system palette.
- Use an odd number of pixels along each side.
  - Provides center pixel around which to focus design.
- Minimum sizes for easy selection:
  - With stylus or pen: 15 pixels square.
  - With mouse: 20 pixels square.
  - With finger: 40 pixels square.

- Provide as large a hot zone as possible.

### **Choosing Icon Images**

- Use existing icons when available.
- Use images for nouns, not verbs.
- Use traditional images.
- Consider user cultural and social norms.

### **Creating Icon Images**

- Create familiar and concrete shapes.
- Create visually and conceptually distinct shapes.
  - Incorporate unique features of an object.
  - Do not display within a border.
- Clearly reflect objects represented.
- Simply reflect objects represented, avoiding excessive detail.
- Create as a set, communicating relationships to one another through common shapes.
- Provide consistency in icon type.
- Create shapes of the proper emotional tone.

### **Drawing Icon Images**

- Provide consistency in shape over varying sizes.
- Do not use triangular arrows in design to avoid confusion with other system symbols.
- When icons are used to reflect varying attributes, express these attributes as meaningfully as possible.
- Provide proper scale and orientation.
- Use perspective and dimension whenever possible.
- Accompany icon with a label to assure intended meaning.

### **Icon Animation and Audition**

- Animation:

— Use:

- To provide feedback.
- For visual interest.

— Make it interruptible or independent of user's primary interaction.

— Do not use it for decoration.

— Permit it to be turned off by the user.

— For fluid animation, present images at 16 or more frames per second.

- Audition:

— Consider auditory icons.

### **The Icon Design Process**

- Define the icon's purpose and use.
- Collect, evaluate, and sketch ideas.
- Draw in black and white.



- Draw using an icon-editing utility or drawing package.
- Test for user
  - Expectations.
  - Recognition.
  - Learning.
- Test for legibility.
- Register new icons in the system's registry.

## **Multimedia**

- The graphical flexibility of the Web permits inclusion of other media on a screen, including images, photographs, video, diagrams, drawings, and spoken audio.
- Multimedia can hold the user's attention, add interest to a screen, entertain, and quickly convey information that is more difficult to present textually.

### **Graphics**

- Use graphics to
  - Supplement the textual content, not as a substitute for it.
  - Convey information that can't be effectively accomplished using text.
  - Enhance navigation through
    - Presenting a site overview.
    - Identifying site pages.
    - Identifying content areas.
- Limit the use of graphics that take a long time to load.
- Coordinate the graphics with all other page elements.
  - Graphics should not look like gratuitous decorations or banner ads.

Graphics contained in Web pages serve several distinct purposes, which can be classified as follows:

- **Navigational.** To identify links that may be followed.
- **Representational.** To illustrate items mentioned in the text.
- **Organizational.** To depict relationships among items mentioned in text.
- **Explanative.** To show how things or processes work.
- **Decorative.** To provide visual appeal and emphasis.

### **Images**

- Ensure all images convey their intended messages.
- General:
  - Use standard images.
  - Emulate real-world objects.
  - Use images consistently.
  - Produce legible images.
  - Provide descriptive text or labels with all images.
  - Distinguish navigational images from decorative images.
  - Minimize
    - The number of presented images.
    - The size of presented images.
    - Restrict single images to 5KB.

- Restrict page images to 20KB.
- Provide thumbnail size images.
- Image animation.
  - Avoid extraneous or gratuitous images.
- Color:
  - Minimize the number of colors in an image.
- Format:
  - Produce images in the most appropriate format.
    - GIF.
    - JPEG.
- Internationalization:
  - Provide for image internationalization.
- Design:
  - Limit large images above the page fold.
  - Use simple background images.
  - Reuse images on multiple pages.

### **Image Maps**

- Use:
  - To provide navigation links to other content.
- Advantages:
  - Can be arrayed in a meaningful and obvious structure.
  - Faster to load than separate images.
- Disadvantages:
  - Consume a significant amount of screen space.
  - Hot spots not always obvious.
  - One's location within image map is not always obvious.
- Guidelines:
  - Use with caution.
  - Provide effective visual cues and emphasis to make it easy to identify link boundaries.
  - Ensure image maps are accessible to the vision-impaired.

### **Photographs/Pictures**

- Use:
  - When every aspect of the image is relevant.
- Guidelines:
  - Use JPEG format.
  - On the initial page, display a small version.
  - Display a thumbnail size image.
  - Zoom in on most relevant detail.
  - Link to larger photos showing as much detail as needed.
  - Include fewer people and objects in less complicated settings than in photos for print.
  - Emphasize close-up shots with clean backgrounds.

### **Video**

- Uses:
  - To show things that move or change over time.
  - To show the proper way to perform a task.

- To show events that cannot be seen directly.
- To convey human behavior and emotions.
  - To provide a personal message.
  - To grab attention.
- Guidelines:
  - Never automatically download a video into a page.
  - Create short segments.
  - Provide controls, including those for playing, pausing, and stopping.
  - Consider using
    - Existing video.
    - Audio only.
    - A slide show with audio.

## **Coloring**

What is perceived as actual color results from the stimulation of the proper receptor in the eye by a received light wave.

### **RGB**

Many color monitors use the three primary colors of light, in various combinations, to create the many colors we see on screens.

### **HSV**

Some palette editors use a convention based on the Munsell method of color notation called HSV, for hue, saturation, and value

### **Dithering**

If pixels of different colors are placed next to each other, this tremor combines the two colors into a third color. This is referred to as *dithering*, and sometimes *texture mapping*

### **Color Uses**

- Use color to assist in formatting a screen
  - Relating or tying elements into groupings.
  - Breaking apart separate groupings of information.
  - Associating information that is widely separated on the screen.
  - Highlighting or calling attention to important information by setting it off from the other information.
- Use color as a visual code to identify
  - Screen components.
  - The logical structure of ideas, processes, or sequences.
  - Sources of information.
  - Status of information.
- Use color to
  - Realistically portray natural objects.
  - Increase screen appeal.
    - Possible Problems with Color
    - High Attention-Getting Capacity
    - Interference with Use of Other Screens

- Varying Sensitivity of the Eye to Different Colors
- Color-Viewing Deficiencies
- Color Connotations

### **Choosing Colors for Categories of Information**

■ Choosing colors for categories of information requires a clear understanding of how the information will be used. Some examples:

- If different parts of the screen are attended to separately, color-code the different parts to focus selective attention on each in turn.
- If decisions are made based on the status of certain types of information on the screen, color-code the types of status that the information may possess.
- If screen searching is performed to locate information of a particular kind or quality, color-code these kinds or qualities for contrast.
- If the sequence of information use is constrained or ordered, use color to identify the sequence.
- If the information displayed on a screen is packed or crowded, use color to provide visual groupings.

■ Use color as a redundant screen code.

### **Emphasis**

■ To draw attention or to emphasize elements, use bright or highlighted colors. To deemphasize elements, use less bright colors.

— The perceived brightness of colors from most to least is white, yellow, green, red, blue.

■ To emphasize separation, use contrasting colors.

— Red and green, blue and yellow.

■ To convey similarity, use similar colors.

— Orange and yellow, blue and violet.

### **Common Meanings**

■ To indicate that actions are necessary, use warm colors.

— Red, orange, yellow.

■ To provide status or background information, use cool colors.

— Green, blue, violet, purple.

■ Conform to human expectations.

— In the job.

— In the world at large.

### **Foregrounds and Backgrounds**

■ Foregrounds:

— Use colors that highly contrast with the background color.

— For text or data, use

• Black.

• Desaturated or spectrum center colors such as white, yellow, or green.

• Warmer more active colors.

— Use colors that possess the same saturation and lightness.

— To emphasize an element, highlight it in a light value of the foreground color, pure white, or yellow.

— To deemphasize an element, lowlight it in a dark value of the foreground color.

#### ■ Backgrounds:

- Use a background color to organize a group of elements into a unified whole.
- Use colors that do not compete with the foreground.
- Use
  - Light-colored backgrounds of low intensity: off-white or light gray.
  - Desaturated colors.
  - Cool, dark colors such as blue or black.
  - Colors on the spectral extremes.

### Physical Impressions

#### ■ Size:

- To convey an impression of
  - Larger: Use bright or saturated colors.
  - Smaller: Use dark or desaturated colors.
  - Similar: Use colors of equal lightness.

#### ■ Weight:

- To convey an impression of
  - Heavy: Use dark, saturated colors.
  - Light: Use light, desaturated colors.

#### ■ Distance:

- To convey an impression of
  - Close: Use saturated, bright, long-wavelength (red) colors.
  - Far: Use saturated, dark, short-wavelength (blue) colors.

#### ■ Height:

- To convey an impression of height, use desaturated, light colors.

#### ■ Depth:

- To convey an impression of depth, use saturated, dark colors.

#### ■ Concentration level:

- To convey an impression of concentration level, use
  - High: Saturated colors.
  - Low: Desaturated colors.

#### ■ Magnitude of change:

- To convey an impression of magnitude of change, use
  - Lowest: Short-wavelength (blue) colors.
  - Highest: Long-wavelength (red) colors.

### Choosing Colors for Web Pages

#### ■ Purpose:

- Color must always have a meaningful purpose.

#### ■ Palette:

- Use the browser 216-color palette.

#### ■ Presentation:

- Minimize the number of presented colors.
- Always consider color in context.
- Use similar or the same color schemes throughout.

- For foregrounds: Use black or strong colors for text and headings.
- For backgrounds: Use weaker contrasting colors such as off-white or light gray.
- Use a uniform color in large areas.
- The smaller the element, the more contrast is required between it and its background.
- Larger images should use
  - Flat, Web-safe colors.
  - Fewer colors than small images.
- Select colors that can be easily reproduced in black and white
- Links:
  - Use default colors for links.
  - Make unselected/unvisited links blue.
  - Make selected/visited links purple.
- Do not display non-link text in link colors.
- Testing:
  - Test all colors.

### **Uses of Color to Avoid**

- Relying exclusively on color.
- Too many colors at one time.
- Highly saturated, spectrally extreme colors together:
  - Red and blue, yellow and purple.
- Low-brightness colors for extended viewing or older viewers.
- Colors of equal brightness.
- Colors lacking contrast:
  - For example, yellow and white; black and brown; reds, blues, and browns against a light background.
- Fully saturated colors for text or other frequently-read screen components.
- Pure blue for text, thin lines, and small shapes.
- Colors in small areas.
- Color for fine details.
- Non-opponent colors.
- Red and green in the periphery of large-scale displays.
- Adjacent colors that only differ in the amount of blue they possess.
- Single-color distinctions for color-deficient users.
- Using colors in unexpected ways.
- Using color to improve legibility of densely packed text.

## **UNIT – V**

### **Prototypes**

A simple, incomplete model or mock-up of a design, is primarily a vehicle for exploration, communication, and evaluation. Its purpose is to obtain user input in design, and to provide feedback to designers.

#### **Hand Sketches and Scenarios**

##### ■ Description:

- Screen sketches created by hand or a drawing package.
- Focus is on the design, not the interface mechanics.
- A low-fidelity prototype.

##### ■ Advantages:

- Can be used very early in the development process.
- Suited for use by entire design team.
- No large investment of time and cost.
- No programming skill needed.
- Easily portable.
- Fast to modify and iterate.
- A rough approximation often yields more substantive critical comments.
- Easier to comprehend than functional specifications.
- Can be used to define requirements.

##### ■ Disadvantages:

- Only a rough approximation.
- Limited in providing an understanding of navigation and flow.
- A demonstration, not an exercise.
- Driven by a facilitator, not the user.
- Limited usefulness for a usability test.
- A poor detailed specification for writing the code.
- Usually restricted to most common tasks.

#### ***Sketch Creation Process***

##### ■ Sketch (storyboard) the screens while determining

- The source of the screen's information.
- The content and structure of individual screens.
- The overall order of screens and windows.

##### ■ Use an erasable medium.

##### ■ Sketch the screens needed to complete each workflow task.

##### ■ Try out selected metaphors and change them as necessary.

##### ■ Sketch common/critical/frequent scenarios first.

- Follow them from beginning to end.
- Go back and build in exceptions.

##### ■ Don't get too detailed; exact control positioning is not important, just overall order

and flow.

- Sketch storyboard as a team, including at least one user.
- Develop online prototypes only when everyone agrees that a complete set of screens has been satisfactorily sketched.

### **Interactive Paper Prototypes**

#### ■ Description:

- Interface components (menus, windows, and screens) constructed of common paper technologies (Post-it notes, transparencies, and so on).
- The components are manually manipulated to reflect the dynamics of the software.
- A low-fidelity prototype.

#### ■ Advantages:

- More illustrative of program dynamics than sketches.
- Can be used to demonstrate the interaction.
- Otherwise, generally the same as for hand-drawn sketches and scenarios.

#### ■ Disadvantages:

- Only a rough approximation.
- A demonstration, not an exercise.
- Driven by a facilitator, not the user.
- Limited usefulness for usability testing.

### **Programmed Facades**

#### ■ Description:

- Examples of finished dialogs and screens for some important aspects of the system.
- Created by prototyping tools.
- Medium-fidelity to high-fidelity prototypes.

#### ■ Advantages:

- Provide a good detailed specification for writing code.
- A vehicle for data collection.

#### ■ Disadvantages:

- May solidify the design too soon.
- May create the false expectation that the “real thing” is only a short time away.
- More expensive to develop.
- More time-consuming to create.
- Not effective for requirements gathering.
- Not all of the functions demonstrated may be used because of cost, schedule limitations, or lack of user interest.
- Not practical for investigating more than two or three approaches.

### **Prototype-Oriented Languages**

#### ■ Description:

- An example of finished dialogs and screens for some important aspects of the



system.

— Created through programming languages that support the actual programming process.

— A high-fidelity prototype.

■ Advantages:

— May include the final code.

— Otherwise, generally the same as those of programmed facades.

■ Disadvantages:

— Generally the same as for programmed facades.

## **Kinds of Tests**

A test is a tool that is used to measure something. The “something” may be

■ Conformance with a requirement.

■ Conformance with guidelines for good design.

■ Identification of design problems.

■ Ease of system learning.

■ Retention of learning over time.

■ Speed of task completion.

## **Guidelines and Standards Review**

■ Description:

— A review of the interface in terms of

• An organization’s standards, design guidelines, or style guide.

• A government standard.

■ Advantages:

— Can be performed by developers.

— Low cost.

— Can identify general and recurring problems.

— Particularly useful for identifying screen design and layout problems.

■ Disadvantages- May miss severe conceptual, navigation, and operational problems.

## **Heuristic Evaluation**

■ Description:

— A detailed evaluation of a system by interface design specialists to identify problems.

■ Advantages:

— Easy to do.

— Relatively low cost.

— Does not waste user’s time.

— Can identify many problems.

■ Disadvantages:

— Evaluators must possess interface design expertise.

— Evaluators may not possess an adequate understanding of the tasks and user communities.

— Difficult to identify systemwide structural problems.

— Difficult to uncover missing exits and interface elements.

- Difficult to identify the most important problems among all problems uncovered.
- Does not provide any systematic way to generate solutions to the problems uncovered.

■ Guidelines:

- Use three to five expert evaluators.
- Choose knowledgeable people
  - Familiar with the project situation.
  - Possessing a long-term relationship with the organization.

**Description.** In a *heuristic evaluation*, interface specialists study a system

### Cognitive Walk-Throughs

■ Description:

- Reviews of the interface in the context of tasks users perform.

■ Advantages:

- Allow a clear evaluation of the task flow early in the design process.
- Do not require a functioning prototype.
- Low cost.
- Can be used to evaluate alternate solutions.
- Can be performed by developers.
- More structured than a heuristic evaluation.
- Useful for assessing exploratory learning.

■ Disadvantages:

- Tedious to perform.
- May miss inconsistencies and general and recurring problems.

■ Guidelines:

- Needed to conduct the walk-through are
  - A general description of proposed system users and what relevant knowledge they possess.
  - A specific description of one or more core or representative tasks to be performed.
  - A list of the correct actions required to complete each of the tasks.
- Review:
  - Several core or representative tasks across a range of functions.
  - Proposed tasks of particular concern.
- Developers must be assigned roles of
  - Scribe to record results of the action.
  - Facilitator to keep the evaluation moving.
- Start with simple tasks.
- Don't get bogged down demanding solutions.
- Limit session to 60 to 90 minutes.

### Think-Aloud Evaluations

■ Description:

- Users perform specific tasks while thinking out loud.
- Comments are recorded and analyzed.

■ Advantages:

- Utilizes actual representative tasks.
- Provides insights into the user's reasoning.
- Helps users focus and concentrate during the session.
- Disadvantages:
  - Some participants may find thinking aloud distracting and unnatural.
  - Can slow the participants thought processes.
  - Can be exhausting for the participant in a long session.
- Guidelines:
  - Develop
    - Several core or representative tasks.
    - Tasks of particular concern.
  - Limit session to 60 to 90 minutes.

### Usability Test

- Description:
  - An interface evaluation under real-world or controlled conditions.
  - Measures of performance are derived for specific tasks.
  - Problems are identified.
- Advantages:
  - Utilizes an actual work environment.
  - Identifies serious or recurring problems.
- Disadvantages:
  - High cost for establishing facility.
  - Requires a test conductor with user interface expertise.
  - Emphasizes first-time system usage.
  - Poorly suited for detecting inconsistency problems.

### Classic Experiments

- Description:
  - An objective comparison of two or more prototypes identical in all aspects except for one design issue.
- Advantages:
  - Objective measures of performance are obtained.
  - Subjective measures of user satisfaction may be obtained.
- Disadvantages:
  - Requires a rigorously controlled experiment to conduct the evaluation.
  - The experiment conductor must have expertise in setting up, running, and analyzing the data collected.
  - Requires creation of multiple prototypes.
- Guidelines:
  - State a clear and testable hypothesis.
  - Specify a small number of independent variables to be manipulated.
  - Carefully choose the measurements.
  - Judiciously select study participants and carefully or randomly assign them to groups.

- Control for biasing factors.
- Collect the data in a controlled environment.
- Apply statistical methods to data analysis.
- Resolve the problem that led to conducting the experiment.

### **Focus Groups**

#### ■ Description:

- A discussion with users about interface design prototypes or tasks.

#### ■ Advantages:

- Useful for
  - Obtaining initial user thoughts.
  - Trying out ideas.
- Easy to set up and run.
- Low cost.

#### ■ Disadvantages:

- Requires experienced moderator.
- Not useful for establishing
  - How people really work.
  - What kinds of usability problems people have.

#### ■ Guidelines:

- Restrict group size to 8 to 12.
- Limit to 90 to 120 minutes in length.
- Record session for later detailed analysis.

### **The Test Plan**

- Define the scope of the test.
- Define the purpose of the test.
  - Performance goals.
  - What the test is intended to accomplish.
- Create a test timetable.
- Define the test methodology.
  - Type of test to be performed.
  - Test limitations.
  - Developer participants.
- Develop scenarios to satisfy the test's purpose.
- Select test participants.
- Identify and schedule the test facility or location.
- Run a pilot test.

### **Test Participants**

- Assemble the proper people to participate in the test.
- Consider allowing users to work in pairs.
- Select the proper number of participants.
- Consider providing compensation or incentives.

### **Usability Test Guidelines**

- Before starting the test,
  - Ensure that all necessary test materials and equipment are available.
  - Explain that the objective is to test the software, not the participants.
  - Explain how the test materials and records will be used.
  - If a consent agreement is to be signed, explain all information on it.
  - If verbal protocols will be collected, let participants practice thinking aloud.
  - Ensure that all participants' questions are answered and that participants are comfortable with all procedures.
- During the test,
  - Minimize the number of people who will interact with the participants.
  - If observers will be in the room, limit them to two or three.
  - Provide a checklist for recording
    - Times to perform tasks.
    - Errors made in performing tasks.
    - Unexpected user actions.
    - System features used/not used.
    - Difficult/easy-to-use features.
    - System bugs or failures.
  - Record techniques and search patterns that participants employ when attempting to work through a difficulty.
  - If participants are thinking aloud, record assumptions and inferences being made.
  - Record the session with a tape recorder or video camera.
  - Do not interrupt participants unless absolutely necessary.
  - If participants need help, provide some response.
    - Provide encouragement or hints.
    - Give general hints before specific hints.
    - Record the number of hints given.
  - Watch carefully for signs of stress in participants:
    - Sitting for long times doing nothing.
    - Blaming themselves for problems.
    - Flipping through documentation without really reading it.
  - Provide short breaks when needed.
  - Maintain a positive attitude, no matter what happens.
- After the test,
  - Hold a post-session interview with participants.
  - Consider using a questionnaire to collect interview data.
  - Tell participants what has been learned in the test.
  - Provide a follow-up questionnaire that asks participants to evaluate the product or tasks performed, if necessary.
  - If videotaping, use tapes only in proper ways.
    - Respect participants' privacy,
    - Get written permission to use tapes.

## **Retest**

- Compile the data from all test participants.

- List the problems the participants had.
- Sort the problems by priority and frequency.
- Develop solutions for the problems.
- Modify the prototype as necessary.
- Test the system again, and again.

### **Evaluate the Working System**

- Collect information on actual system usage through
  - Interviews and focus group discussions.
  - Surveys.
  - Support line.
  - Online suggestion box or trouble reporting.
  - Online bulletin board.
  - User newsletters and conferences.
  - User performance data logging.
- Respond to users who provide feedback.

## **Hypermedia**

- **Hypermedia** is used as a logical extension of the term hypertext in which graphics, audio, video, plain text and hyperlinks intertwine to create a generally non-linear medium of information.
- This contrasts with the broader term *multimedia*, which may be used to describe non-interactive linear presentations as well as hypermedia.
- Hypermedia should not be confused with hypergraphics or super-writing which is not a related subject. It is also related to the field of Electronic literature. A term first used in a 1965 article by Ted Nelson.
- The World Wide Web is a classic example of hypermedia, whereas a non-interactive cinema presentation is an example of standard multimedia due to the absence of hyperlinks.

### **Hypermedia development tools**

- Hypermedia may be developed a number of ways. Any programming tool can be used to write programs that link data from internal variables and nodes for external data files.
- Multimedia development software such as Adobe Flash, Adobe Director, Macromedia Authorware, and MatchWare Mediator may be used to create stand-alone hypermedia applications, with emphasis on entertainment content.
- Some database software such as Visual FoxPro and FileMaker Developer may be used to develop stand-alone hypermedia applications, with emphasis on educational and business content management.
- Hypermedia applications may be developed on embedded devices for the mobile and the Digital signage industries using the Scalable Vector Graphics (SVG) specification from W3C (World Wide Web Consortium).

- Software applications such as [Ikivo Animator](#) and [Inkscape](#) simplify the development of Hypermedia content based on SVG. Embedded devices such as [iPhone](#) natively support SVG specifications and may be used to create mobile and distributed Hypermedia applications.
- Hyperlinks may also be added to [data files](#) using most [business software](#) via the limited [scripting](#) and [hyperlinking](#) features built in.
- Documentation software such as the [Microsoft Office Suite](#) allows for hypertext links to other content within the same file, other external files, and [URL](#) links to files on external [file servers](#).
- For more emphasis on graphics and [page layout](#), hyperlinks may be added using most modern [desktop publishing](#) tools. This includes [presentation programs](#), such as [Microsoft Powerpoint](#), add-ons to print layout programs such as [Quark Immedia](#), and tools to include hyperlinks in [PDF](#) documents such as [Adobe InDesign](#) for creating and [Adobe Acrobat](#) for editing.
- [Hyper Publish](#) is a tool specifically designed and optimized for hypermedia and hypertext management. Any [HTML Editor](#) may be used to build [HTML](#) files, accessible by any web browser.
- [CD/DVD authoring](#) tools such as [DVD Studio Pro](#) may be used to hyperlink the content of [DVDs](#) for [DVD players](#) or web links when the disc is played on a [personal computer](#) connected to the internet.

## Visualization

- **Visualization** is any technique for creating [images](#), [diagrams](#), or [animations](#) to communicate a message.
- Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of man.
- Examples from history include [cave paintings](#), [Egyptian hieroglyphs](#), Greek [geometry](#), and [Leonardo da Vinci's](#) revolutionary methods of technical drawing for engineering and scientific purposes.
- Visualization today has ever-expanding applications in science, education, engineering (e.g. product visualization), [interactive multimedia](#), [medicine](#), etc.
- Typical of a visualization application is the field of [computer graphics](#).
- The invention of computer graphics may be the most important development in visualization since the invention of [central perspective](#) in the [Renaissance](#) period. The development of [animation](#) also helped advance visualization.

## Fields of visualization

- A scientific visualization of an extremely large simulation of a Raleigh-Taylor instability caused by two mixing fluids.
- As a subject in computer science, **data visualization** or scientific visualization is the use of interactive, sensory representations, typically visual, of abstract data to reinforce cognition, hypothesis building and reasoning.

## Educational visualization

- Educational visualization is using a simulation normally created on a computer to create an image of something so it can be taught about. In the Roman times, this is very useful when teaching about a topic which is difficult to otherwise see, for example, atomic structure, because atoms are far too small to be studied easily without expensive and difficult to use scientific equipment.
- It can also be used to view past events, such as looking at dinosaurs, or looking at things that are difficult or fragile to look at in reality like the human skeleton, without causing physical or mental harm to a subjective volunteer or cadaver.

## Information visualization

- Information visualization concentrates on the use of computer-supported tools to explore large amount of abstract data.
- The term "information visualization" was originally coined by the User Interface Research Group at Xerox PARC and included Dr. Jock Mackinlay. Practical application of information visualization in computer programs involves selecting, transforming and representing abstract data in a form that facilitates human interaction for exploration and understanding.
- Important aspects of information visualization are dynamics of visual representation and the interactivity. Strong techniques enable the user to modify the visualization in real-time, thus affording unparalleled perception of patterns and structural relations in the abstract data in question.

## Knowledge visualization

- The use of visual representations to transfer knowledge between at least two persons aims to improve the transfer of knowledge by using computer and non-computer based visualization methods complementarily.
- Examples of such visual formats are sketches, diagrams, images, objects, interactive visualizations, information visualization applications and imaginary visualizations as in stories. While information visualization concentrates on the use of computer-supported



tools to derive new insights, knowledge visualization focuses on transferring insights and creating new knowledge in groups.

- Beyond the mere transfer of facts, knowledge visualization aims to further transfer insights, experiences, attitudes, values, expectations, perspectives, opinions, and predictions by using various complementary visualizations.

### Product Visualization

- Product Visualization involves visualization software technology for the viewing and manipulation of 3D models, technical drawing and other related documentation of manufactured components and large assemblies of products.
- It is a key part of Product Lifecycle Management. Product visualization software typically provides high levels of photorealism so that a product can be viewed before it is actually manufactured. This supports functions ranging from design and styling to sales and marketing.
- *Technical visualization* is an important aspect of product development. Originally technical drawings were made by hand, but with the rise of advanced computer graphics the drawing board has been replaced by computer-aided design (CAD).
- CAD-drawings and models have several advantages over hand-made drawings such as the possibility of 3-D modeling, rapid prototyping and simulation.

### Visual communication

- Visual communication is the communication of ideas through the visual display of information. Primarily associated with two dimensional images, it includes: alphanumerics, art, signs, and electronic resources.
- Recent research in the field has focused on web design and graphically oriented usability.

### Visual analytics

- Visual analytics focuses on human interaction with visualization systems as part of a larger process of data analysis. Visual analytics has been defined as "the science of analytical reasoning supported by the interactive visual interface" <sup>[2]</sup>.
- Its focus is on human information discourse (interaction) within massive, dynamically changing information spaces. Visual analytics research concentrates on support for perceptual and cognitive operations that enable users to detect the expected and discover the unexpected in complex information space.
- Technologies resulting from visual analytics find their application in almost all fields, but are being driven by critical needs (and funding) in biology and national security.

### Visualization techniques

The following are examples of some common visualization techniques:

- Constructing isosurfaces
- direct volume rendering
- Streamlines, streaklines, and pathlines
- table, matrix
- charts (pie chart, bar chart, histogram, function graph, scatter plot, etc.)
- graphs (tree diagram, network diagram, flowchart, existential graph, etc.)
- Maps
- parallel coordinates - a visualization technique aimed at multidimensional data
- treemap - a visualization technique aimed at hierarchical data
- Venn diagram
- Euler diagram
- Chernoff face
- Hyperbolic trees

## **WWW**

The **World Wide Web** (commonly shortened to **the Web**) is a system of interlinked hypertext documents accessed via the Internet. With a Web browser, a user views Web pages that may contain text, images, videos, and other multimedia and navigates between them using hyperlinks.

The World Wide Web was created in 1989 by British scientist Sir Tim Berners-Lee, working at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland, and released in 1992.

Since then, Berners-Lee has played an active role in guiding the development of Web standards (such as the markup languages in which Web pages are composed), and in recent years has advocated his vision of a Semantic Web.

### **[edit] How it works**

Viewing a Web page on the World Wide Web normally begins either by typing the URL of the page into a Web browser, or by following a hyperlink to that page or resource. The Web browser then initiates a series of communication messages, behind the scenes, in order to fetch and display it.

First, the server-name portion of the URL is resolved into an IP address using the global, distributed Internet database known as the domain name system, or DNS. This IP address is necessary to contact and send data packets to the Web server.

The browser then requests the resource by sending an HTTP request to the Web server at that particular address.

In the case of a typical Web page, the HTML text of the page is requested first and parsed immediately by the Web browser, which will then make additional requests for images and any other files that form a part of the page.

Statistics measuring a website's popularity are usually based on the number of '[page views](#)' or associated server '[hits](#)', or file requests, which take place.

Having received the required files from the Web server, the browser then [renders](#) the page onto the screen as specified by its HTML, [CSS](#), and other Web languages. Any images and other resources are incorporated to produce the on-screen Web page that the user sees.

## Standards

Many formal standards and other technical specifications define the operation of different aspects of the World Wide Web, the Internet, and computer information exchange.

Many of the documents are the work of the [World Wide Web Consortium](#) (W3C), headed by Berners-Lee, but some are produced by the [Internet Engineering Task Force](#) (IETF) and other organizations.

Usually, when Web standards are discussed, the following publications are seen as foundational:

- Recommendations for [markup languages](#), especially [HTML](#) and [XHTML](#), from the W3C. These define the structure and interpretation of [hypertext](#) documents.
- Recommendations for [stylesheets](#), especially [CSS](#), from the W3C.
- Standards for [ECMAScript](#) (usually in the form of [JavaScript](#)), from [Ecma International](#).
- Recommendations for the [Document Object Model](#), from W3C.

Additional publications provide definitions of other essential technologies for the World Wide Web, including, but not limited to, the following:

- *Uniform Resource Identifier (URI)*, which is a universal system for referencing resources on the Internet, such as hypertext documents and images. URIs, often called URLs, are defined by the IETF's [RFC 3986](#) / STD 66: *Uniform Resource Identifier (URI): Generic Syntax*, as well as its predecessors and numerous [URI scheme-defining RFCs](#);
- *HyperText Transfer Protocol (HTTP)*, especially as defined by [RFC 2616: HTTP/1.1](#) and [RFC 2617: HTTP Authentication](#), which specify how the browser and server authenticate each other.

## Java

A significant advance in Web technology was [Sun Microsystems'](#) [Java platform](#). It enables Web pages to embed small programs (called [applets](#)) directly into the view. These applets run on the end-user's computer, providing a richer user interface than simple Web pages.

Java client-side applets never gained the popularity that Sun had hoped for a variety of reasons, including lack of integration with other content (applets were confined to small boxes within the

rendered page) and the fact that many computers at the time were supplied to end users without a suitably installed Java Virtual Machine, and so required a download by the user before applets would appear.

Adobe Flash now performs many of the functions that were originally envisioned for Java applets, including the playing of video content, animation, and some rich GUI features. Java itself has become more widely used as a platform and language for server-side and other programming.

## JavaScript

JavaScript, on the other hand, is a scripting language that was initially developed for use within Web pages. The standardized version is ECMAScript.

While its name is similar to Java, JavaScript was developed by Netscape and has very little to do with Java, although the syntax of both languages is derived from the C programming language.

In conjunction with a Web page's Document Object Model (DOM), JavaScript has become a much more powerful technology than its creators originally envisioned. <sup>[citation needed]</sup> The manipulation of a page's DOM after the page is delivered to the client has been called Dynamic HTML (DHTML), to emphasize a shift away from *static* HTML displays.

In simple cases, all the optional information and actions available on a JavaScript-enhanced Web page will have been downloaded when the page was first delivered.

Ajax ("Asynchronous JavaScript and XML") is a group of interrelated web development techniques used for creating interactive web applications that provide a method whereby parts *within* a Web page may be updated, using new information obtained over the network at a later time in response to user actions.

This allows the page to be more responsive, interactive and interesting, without the user having to wait for whole-page reloads. Ajax is seen as an important aspect of what is being called Web 2.0. Examples of Ajax techniques currently in use can be seen in Gmail, Google Maps, and other dynamic Web applications.

## Publishing Web pages

Web page production is available to individuals outside the mass media. In order to publish a Web page, one does not have to go through a publisher or other media institution, and potential readers could be found in all corners of the globe.

Many different kinds of information are available on the Web, and for those who wish to know other societies, cultures, and peoples, it has become easier.

The increased opportunity to publish materials is observable in the countless personal and social networking pages, as well as sites by families, small shops, etc., facilitated by the emergence of free [Web hosting](#) services

### **WWW prefix in Web addresses**

The letters "www" are commonly found at the beginning of [Web addresses](#) because of the long-standing practice of naming Internet hosts (servers) according to the services they provide.

This use of such prefixes is not required by any technical standard; indeed, the first Web server was at "nxoc01.cern.ch",<sup>[32]</sup> and even today many Web sites exist without a "www" prefix. The "www" prefix has no meaning in the way the main Web site is shown. The "www" prefix is simply one choice for a Web site's host name.

Some Web browsers will automatically try adding "www." to the beginning, and possibly ".com" to the end, of typed URLs if no host is found without them. If major web browser will also prefix "http://www." and append ".com" to the address bar contents if the Control and Enter keys are pressed simultaneously.

### **Interface Design Tools**

- Tools used for designing the interface, development environments for writing code, and toolkits of graphical user interfaces.

### **Epark**

- EnhancementPak™ (EPak) is an advanced set of OSF/Motif™ widgets extensively tested in hundreds of large scale commercial software applications.
- EPak provides developers with tested, easy-to-use components that eliminate the time, cost, and risk of custom widget development.
- Key Features:
  - Windows95-like controls
  - Easy to use geometry managers and containers
  - Data presentation, application
  - Binary and source code available

- No royalties or runtime fees

## Converter

- At Integrated Computer Solutions we are committed to your success.
- That's why we offer several products that will let you migrate from software that does not meet your needs to the state-of-the-art, industry standard that will help you retain the value of your code.
- Our code conversion is more than simply translating to UIL, you need to know how to avoid creating more problems than you solve.

## BX PRO

- Builder Xcessory PRO™ (BX PRO) provides C, C++ or Java developers with everything they need to develop and manage GUI projects of any size from start to finish.
- Benefits
  - Speed GUI development
  - Eliminate unmaintainable code
  - Quickly implement changes

## ICE Interface

- ICEfaces is a standards-compliant extension to JavaServer Faces (JSF) for building and deploying rich web applications. ICEfaces™ creates rich user interaction and provides superior presentation characteristics like:
- Smooth, incremental page updates with in-place editing and no full page refresh
- User context preservation during page update, including scrollbar positioning and user focus
- Asynchronous page updates driven from the application in real time
- Fine-grained user interaction during form entry that augments the standard submit/response loop
- ICEfaces is an industrial strength solution that leverages the JSF application framework and the entire J2EE ecosystem of tools and execution environments. Rich application features are developed in pure Java, and in a pure thin-client model.
- There are no Applets, browser plugins, or JavaScript-laden pages thanks to ICEfaces' breakthrough Direct-to-DOM™ rendering technology. And because ICEfaces applications are JSF applications, your J2EE and JSF development skills apply directly with no learning curve.

## **Software Tools**

### **Quick Test Professional (QTP)**

Quick Test Professional (QTP) is an automated functional Graphical User Interface (GUI) testing tool that allows the automation of user actions on a web or client based computer application. It is primarily used for functional regression test automation. QTP uses a scripting language built on top of VBScript to specify the test procedure, and to manipulate the objects and controls of the application under test.

### **WinRunner**

WinRunner, Mercury Interactive's enterprise functional testing tool. It is used to quickly create and run sophisticated automated tests on your application. Winrunner helps you automate the testing process, from test development to execution. You create adaptable and reusable test scripts that challenge the functionality of your application. Prior to a software release, you can run these tests in a single overnight run- enabling you to detect and ensure superior software quality.

### **LoadRunner**

LoadRunner is divided up into 3 smaller applications:

The Virtual User Generator allows us to determine what actions we would like our Vusers, or virtual users, to perform within the application. We create scripts that generate a series of actions, such as logging on, navigating through the application, and exiting the program.

The Controller takes the scripts that we have made and runs them through a schedule that we set up. We tell the Controller how many Vusers to activate, when to activate them, and how to group the Vusers and keep track of them.

The Results and Analysis program gives us all the results of the load test in various forms. It allows us to see summaries of data, as well as the details of the load test for pinpointing problems or bottlenecks.

### **TestDirector**

TestDirector, the industry's first global test management solution, helps organizations deploy high-quality applications more quickly and effectively. Its four modules Requirements, Test



Plan, Test Lab, and Defects are seamlessly integrated, allowing for a smooth information flow between various testing stages. The completely Web-enabled TestDirector supports high levels of communication and collaboration among distributed testing teams, driving a more effective, efficient global application-testing process

## Silk Test

Silk Test is a tool specifically designed for doing **REGRESSION AND FUNCTIONALITY** testing. It is developed by Segue Software Inc. Silk Test is the industry's leading functional testing product for e-business applications, whether Window based, Web, Java, or traditional client/server-based. Silk Test also offers test planning, management, direct database access and validation, the flexible and robust 4Test scripting language, a built in recovery system for unattended testing, and the ability to test across multiple platforms, browsers and technologies.

You have two ways to create automated tests using silktest:

1. Use the **Record Testcase command** to record actions and verification steps as you navigate through the application.
2. Write the testcase manually using the **Visual 4Test scripting language**.

### 1. Record Testcase

The Record / Testcase command is used to record actions and verification steps as you navigate through the application. Tests are recorded in an object-oriented language **called Visual 4Test**. The recorded testreads like a logical trace of all of the steps that were completed by the user. The Silk Test point and click verification system allows you to record the verification step by selecting from a list of properties that are appropriate for the type of object being tested. For example, you can verify the text is stored in a text field.

### 2. Write the Testcase manually

We can write tests that are capable of accomplishing many variations on a test. The key here is re-use. A test case can be designed to take parameters including input data and expected results. This "data-driven" testcase is really an instance of a class of test cases that performs certain steps to drive and verify the application-under-test. Each instance varies by the data that it carries. Since far fewer tests are written with this approach, changes in the GUI will result in reduced effort in updating tests. A data-driven test design also allows for the externalization of testcase



data and makes it possible to divide the responsibilities for developing testing requirements and for developing test automation. For example, it may be that a group of domain experts create the Testplan Detail while another group of test engineers develop tests to satisfy those requirements.

In a script file, an automated testcase ideally addresses one test requirement. Specifically, a 4Test function that begins with the test case keyword and contains a sequence of 4Test statements. It drives an application to the state to be tested, verifies that the application works as expected, and returns the application to its base state.

A script file is a file that contains one or more related testcases. A script file has a .t extension, such as find .t

### **Rational Robot**

Rational Robot is a complete set of components for automating the testing of Microsoft Windows client/server and Internet applications running under Windows NT 4.0, Windows 2000, Windows 98, and Windows 95.

The main component of Robot lets you start recording tests in as few as two mouse clicks. After recording, Robot plays back the tests in a fraction of the time it would take to repeat the actions manually.

Other components of Robot are:

- Rational Administrator Use to create and manage Rational projects, which store your testing information.
- Rational TestManager Log Use to review and analyze test results.
- Object Properties, Text, Grid, and Image Comparators Use to view and analyze the results of verification point playback.
- Rational SiteCheck Use to manage Internet and intranet Web sites.